JSP (Java Sever Page)
- 표현 언어(Expression Language)

표현 언어

- Expression Language (JSTL : JSP Standard Tag Library)
- JSP에서 사용가능한 새로운 스크립트 언어
- EL의 주요 기능
  - JSP의 네 가지 기본 객체가 제공하는 영역의 속성 사용
    -page
    -request
    -session
    -application
  - 집합 객체에 대한 접근 방법 제공
  - 수치 연산, 관계 연산, 논리 연산자 제공
  - 자바 클래스 메서드 호출 기능 제공 beans의 특징을 가져간다.
  - 표현언어만의 기본 객체 제공
- 간단한 구문 때문에 표현식 대신 사용

01. 표현 언어

1. 표현 언어란

- 표현 언어(Expression Language)는 처음 JSTL(JSP Standard Tag Library)이 소개되었을 때 나온 것으로, MVC 패턴에 따라 뷰(view) 역할을 수행하는 JSP를 더욱 효과적으로 만들려는 목적으로 개발되었다.
- 표현 언어는 간단한 방법으로 데이터를 표현하기 위해 고안된 언어인 SPEL(Simplest Possible Expression Language)에 기본을 두고 있다.

JSP 표현식 사용

객체 생성 -> 속성에 저장

```
<H2>
<jsp:useBean id="test" class="TestBean" />
<%= test.getName() %> 혹은 <jsp:getProperty name="test" property="name" />
</H2>
```

EL 사용

```
<H2>
${test.name}
</H2>
```

${test.name}

속성이름 , getName()

01. 표현 언어

request 속성에 data저장 list_view.jsp

- 표현 언어 사용을 위해서는 현재 페이지에서 출력하고자 하는 데이터가 미리 확보 되어 있어야 한다. 예를 들면 page, request, application, session 내장 객체중 하나에 사용하고자 하는 객체가 있어야만 표현언어를 이용해 데이터 출력이 가능 하다.
- 표현언어의 기본적인 문법은 다음과 같다.

- 표현 언어는 '$'로 시작한다.
- 모든 내용은 '{표현식}'과 같이 구성된다.
- 표현식에는 기본적으로 변수 이름, 혹은 '객체_이름.멤버변수_이름'구조로 이루어진다.
- 표현식에는 부가적으로 숫자, 문자열, boolean, null과 같은 상수 값도 올 수 있다.
- 표현식에는 기본적인 연산을 할 수 있다.

메소드 호출

01. 표현 언어

2. 표현 언어에서 사용할 수 있는 내장 객체

page, request, session ,application 에 저장된 객체

- 표현언어에서는 객체가 생성되어 전달된다는 것을 가정하고 있다. 따라서 표현언어 에서 사용 시점에 객체를 선언할 필요가 없음.
- 표연언어에서는 다음과 같이 객체에 접근 할 수 있음.

> ${member.id} 혹은 ${member["id"]} → member 객체의 getId() 메서드 호출과 동일
>
> ${row[0]} → row라는 이름의 컬렉션 객체의 첫 번째 값
>
> + Jstl -> 반복문과 같은기능의 tag

- 또한 몇몇 내장객체를 통해 컨테이너가 제공하는 다른 객체에 접근할 수 있는 방법을 제공하고 있다.

01. 표현 언어

표현 언어에서 사용할 수 있는 내장객체

속성을 저장하는 영역

내장객체	기능
pageScope	page 범위에 포함된 속성 값에 접근할 수 있는 객체다.
requestScope	request 범위에 포함된 속성 값에 접근할 수 있는 객체다.
sessionScope	session 범위에 포함된 속성 값에 접근할 수 있는 객체다.
applicationScope	application 범위에 포함된 속성 값에 접근할 수 있는 객체다.
param ${param.page}	request.getParameter("xxx")로 얻을 수 있는 값들이다. ${param.xxx}처럼 사용한다.
paramValues	request.getParameterValues("xxx")와 동일한 기능을 수행한다. ${paramValues.xxx}처럼 사용한다.
header	request.getHeader("xxx")와 동일한 기능을 수행한다. ${header.xxx}처럼 사용한다.
headerValues	request.getHeaderValues("xxx")와 동일한 기능을 수행한다. ${headerValues.xxx}처럼 사용한다.
initParam	컨텍스트의 초기화 매개변수 값이다.
cookie	쿠키 정보에 접근할 수 있는 객체다.
pageContext	pageContext 객체다.

■ 표현언어에서 기본 객체

useELObject.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%
        request.setAttribute("name", "유영진");
%>
<html>
<head><title>EL Object</title></head>
<body>

요청 URI: ${pageContext.request.requestURI} <br>
request의 name 속성: ${requestScope.name} <br>
code 파라미터: ${param.code}

</body>
</html
```

getRequest()

getRequestURI

■ 표현언어 데이터 타입

- 불리언(Boolean) 타입 - true 와 false
- 정수타입 - 0~9로 이루어진 정수 값.
- 실수타입 - 0~9로 이루어져 있으며, 소수점('.')을 사용할 수 있고, 3.24e3 과 같이 지수형으로 표현 가능
- 문자열 타입 - 따옴표(' 또는 ")로 둘러싼 문자열.
  - 작은 따옴표 사용시, 값에 포함된 작은 따옴표는 ₩'로 입력
  - ₩ 기호 자체는 ₩₩ 로 표시한다.
- 널 타입 - null

■ 표현언어에서 객체에 접근

- ${<표현1>.<표현2>} 형식 사용
- 처리 과정
  1. <표현1>을 <값1>로 변환한다.
  2. <값1>이 null이면 null을 리턴한다.
  3. <값1>이 null이 아닐 경우 <표현2>를 <값2>로 변환한다.
     1. <값2>가 null이면 null을 리턴한다.
  4. <값1>이 Map, List, 배열인 경우
     1. <값1>이 Map이면
        1. <값1>.containsKey(<값2>)가 false이면 null을 리턴한다.
        2. 그렇지 않으면 <값1>.get(<값2>)를 리턴한다.

■ 표현언어에서 객체에 접근

2. <값1>이 List나 배열이면
    1. <값2>가 정수 값인지 검사한다. (정수 값이 아닐 경우 에러 발생)
    2. <값1>.get(<값2>) 또는 Array.get(<값1>, <값2>)를 리턴한다.
    3. 위 코드가 예외를 발생하면 에러를 발생한다.
1. <값1>이 다른 객체이면
    1. <값2>를 문자열로 변환한다.
    2. <값1>이 이름이 <값2>이고 읽기 가능한 프로퍼티를 포함하고 있다면 프로퍼티의 값을 리턴한다.
    3. 그렇지 않을 경우 에러를 발생한다.

3. 표현 언어에서 사용할 수 있는 연산자

– 표현 언어에서는 표현식 부분에서 기본적인 연산을 할 수 있다.

연산자	기능	연산자	기능
+	더하기	-	빼기
*	곱하기	/ or div	나누기
% of mod	몫		

연산자	기능	연산자	기능
== 혹은 eq	같다.	!= 혹은 ne	같지 않다.
< 혹은 lt	좌변이 우변보다 작다.	> 혹은 gt	좌변이 우변보다 크다.
<= 혹은 le	좌변이 우변보다 같거나 작다.	>= 혹은 ge	좌변이 우변보다 같거나 크다.
a?b :c	a가 참이면 b, 거짓이면 c를 반환한다.		

연산자	기능
&& 혹은 and	AND 연산
|| 혹은 or	OR 연산
! 혹은 not	NOT

01. 표현 언어

- empty 연산자
  - empty <값> ${ member empty }
    - 값이 null이면, true
    - 값이 빈 문자열("")이면, true
    - 값의 길이가 0인 배열이나 콜렉션이면 true
    - 이 외의 경우에는 false

- 비교 선택 연산자
  - <수식> ? <값1> : <값2>

02. [기본실습] 표현 언어의 기본 이해

1. 실습

- 실제 표현언어가 활용되는 사례를 통해 표현 언어에 익숙해 지기 위한 간단한 애플리케이션을 개발한다.
- 애플리케이션은 상품 목록을 보여주고 상품을 선택하면 선택된 상품에 대한 정보를 보여주는 예제이다.
  - Product.java : 상품 정보를 제공하는 빈즈 클래스로 jsp에 데이터를 공급하는 역할을 한다.
  - ProductList.jsp : 상목 목록을 출력하는 j네 파일로, Product 클래스로부터 상품 목록을 가져와 출력.
  - ProductSel.jsp : ProductList.jsp에서 상품을 선택하고 <확인> 버튼을 눌렀을때 호출되는 jsp로, 표현 언어를 이용해 데이터를 출력한다.

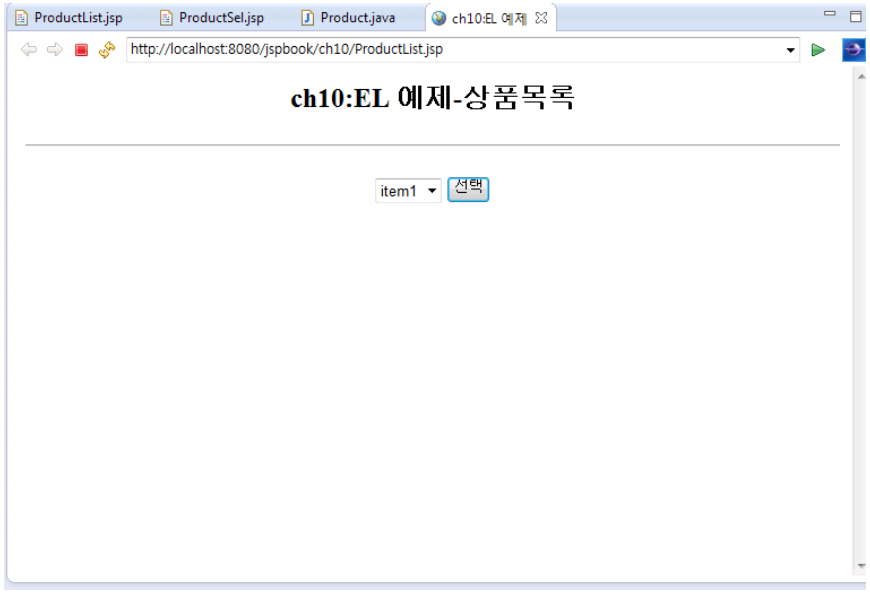
디렉터리	파일
src₩	Product.java
WebContent₩	ProductList.jsp
WebContent₩	ProductSel.jsp

02. [기본실습] 표현 언어의 기본 이해

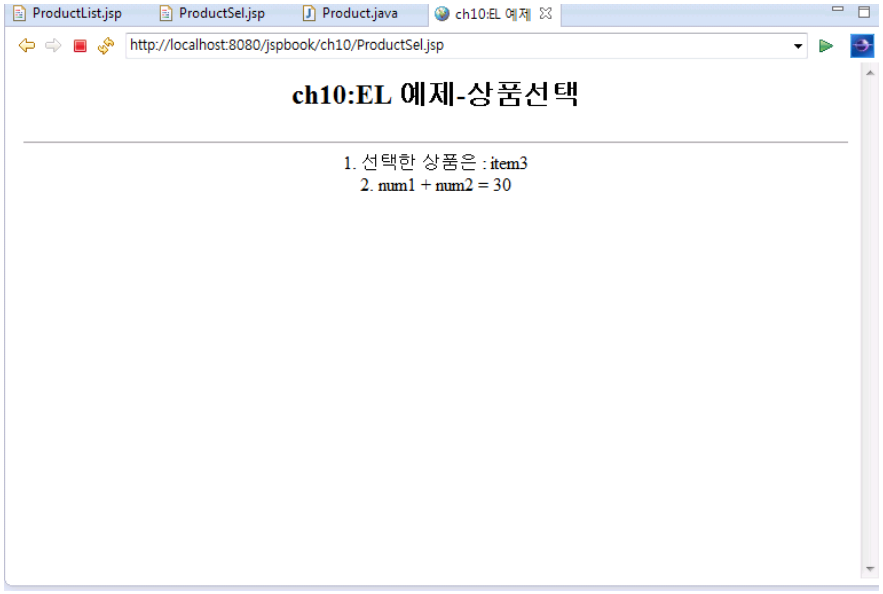
2. 소스 작성

- **Product 클래스 작성 :** Product 클래스(Product.java)
- **ProductList.jsp 작성 :** 상품 목록 출력(ProductList.jsp)
- **ProductSel.jsp 작성 :** 상품 선택(ProductSel.jsp)

3. 실행 및 결과 확인



ProductList.jsp 실행 화면



ProductSel.jsp 실행 화면

```
package jspbook.ch10;
public class Product {
	// 상품 목록을 보관할 배열
	private String[] productList={"item1","item2","item3","item4","item5"};
	// 웹 테스트를 위한 변수값
	private int num1 = 10;
	private int num2 = 20;

	public int getNum1() {
		return num1;
	}
	public int getNum2() {
		return num2;
	}
	public String[] getProductList() {
		return productList;
	}
}
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8 " pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>ch10:EL 예제</title>
</head>
<body>
<H2>ch10:EL 예제-상품목록</H2>
<HR>
<form name=form1 method=POST action=ProductSel.jsp>
        <jsp:useBean id="product" class="jspbook.ch10.Product" scope="session"/>
        <select name="sel">
        <%
                for(String item : product.getProductList()) {
                        out.println("<option>"+item+"</option>");
                }
        %>
        </select>
        <input type="submit" value="선택"/>
</form>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>ch10:EL 예제</title>
</head>
<body>
<div>
<H2>ch10:EL 예제-상품선택</H2>
<HR>
1. 선택한 상품은 : ${param.sel} <BR>
2. num1 + num2 = ${product.num1+product.num2} <BR>
</div>
</body>
</html>
```

02. [기본실습] 표현 언어의 기본 이해

주요 소스코드 분석(Product.java)

- 데이터 관리를 위한 빈즈 클래스
- 상품정보는 편의상 배열로 만들어 제공
- 표현언어 실습을 위해 num1, num2 라는 변수를 미리 설정해 둠.

```
06    private String[] productList = {"item", "item2", "item3", "item4", "item5"};
07    private int num1 = 10;
08    private int num2 = 20;
```

주요 소스코드 분석(ProductList.jsp)

- 상품 목록을 HTML 의 <select><option> 을 이용해 출력하는 JSP
- <jsp:useBean>을 이용해 Product 클래스 사용
- for 문을 이용해 배열 데이터를 <select><option> 문과 조합해 출력

```
13 <jsp:useBean id="product" class="Product" scope="session" />
14    <select name="sel">
15    <%
16        for(String item : product.getProductList()) {
17            out.println("<option>"+item+"</option>");
18        }
19    %>
20    </select>
```

02. [기본실습] 표현 언어의 기본 이해

주요 소스코드 분석(ProductSel.jsp)

- 선택된 상품정보를 출력하는 JSP
- 표현언어를 통해 선택된 상품과 product 객체의 정보를 출력

```
12 1. 선택한 상품은 : ${param.sel} <BR>
13 2. num1 + num2 = ${product.num1+product.num2} <BR>
```

표현언어의 사용법

- request나 session 속성으로 전달한 값을 출력

  결과 -> view 출력

  로그인

- 액션 태그나 커스텀 태그의 속성 값
  - <jsp:include page="/lo/${layout.module}.jsp" flush="true" />

    JSTL

- 함수 호출
  - 코드의 간결함 및 가독성 향상

■ 표현언어의 사용법

- **request나 session 속성으로 전달한 값을 출력**

```
<%
        request.setAttribute(“memberInfo", memberInfo);
%>
<jsp:include page=“<%= includePage%>” flush=“false” />
```

```
<%-- 포함되는 JSP에서 스크립트릿과 표현식  --%>
<%
        MemberInfo memberInfo  =
                        (memberInfo) request.setAttribute(“memberinfo");
%>
이름 : <%= memberinfo.getName() %>
. . .
<%--  --%>
이름 :${memberInfo.name}
```

■ 표현언어의 사용법

- **액션 태그나 커스텀 태그의 속성 값**

```
<jsp:include page="/layout/Top.jsp" flush="false" />
```

```
<jsp:include page='<%= "/layout/+ layout.getMouduleName() +".jsp"%>'
flush="true" />
```

- **표현언어 사용**

```
<jsp:include page="/layout/${layout.getMouduleName}.jsp" flush="true" />
```

```
<jsp:include page="/${folderName}/${layout.getMouduleName}.jsp"
flush="true" />
```

```
<jsp:include
        page='<%= "/" + folderName + "/" + layout.getMouduleName()
+ ".jsp" %>' flush="true" />
```